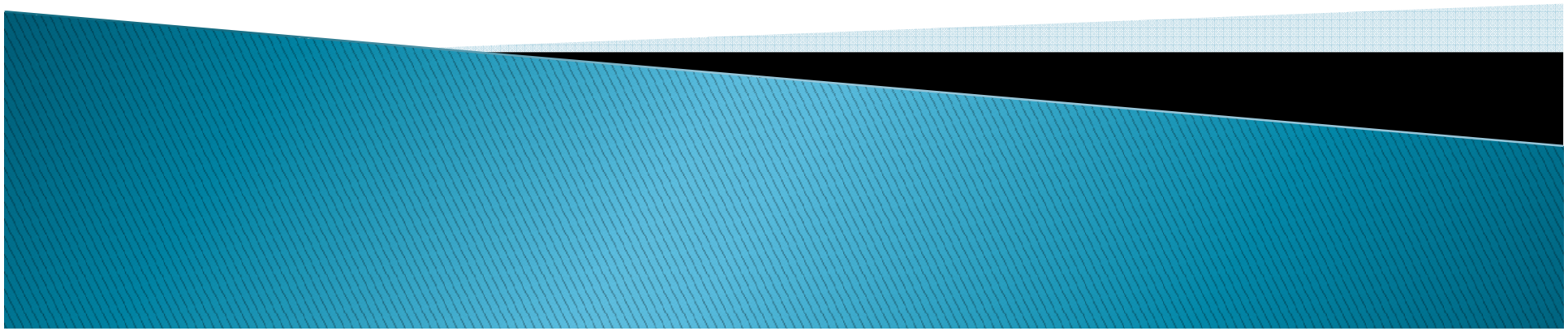
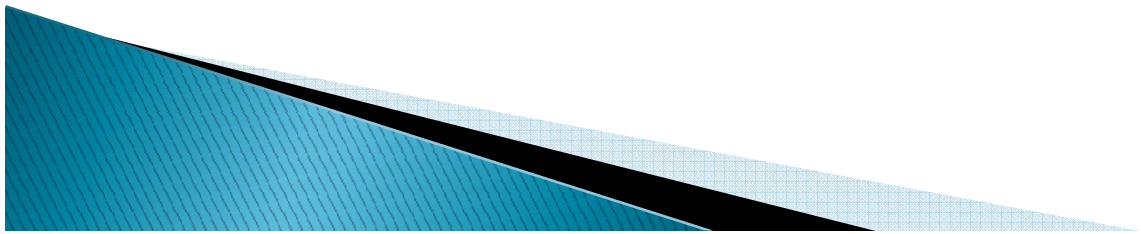
# UML Notations

Madalina Croitoru



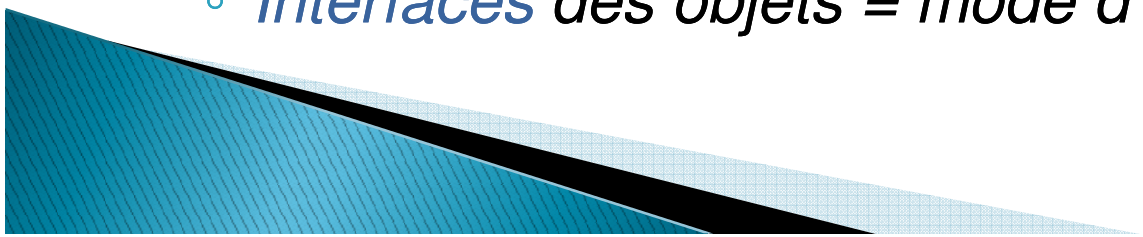
# Systèmes objet

- Les systèmes objets
  - le logiciel est une collection d'objets dissociés définis par des propriétés
  - L'approche objet facilite le développement et l'évolution d'applications complexes



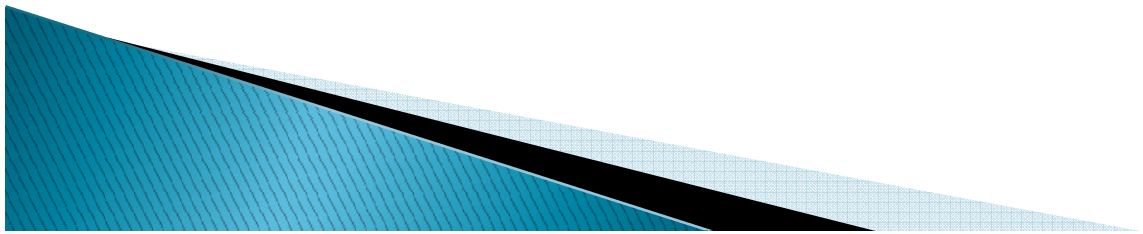
# Méthodologies orientées objet

- Modifiabilité
  - *Du fait de l'encapsulation et de l'abstraction des données*
- Réutilisabilté
  - *Bibliothèque de classes*
  - *Construction d'objets par instanciation et par composition d'objets*
- Lisibilité
  - *Détails d'implémentation cachés*
  - *Interfaces des objets = mode d'emploi*



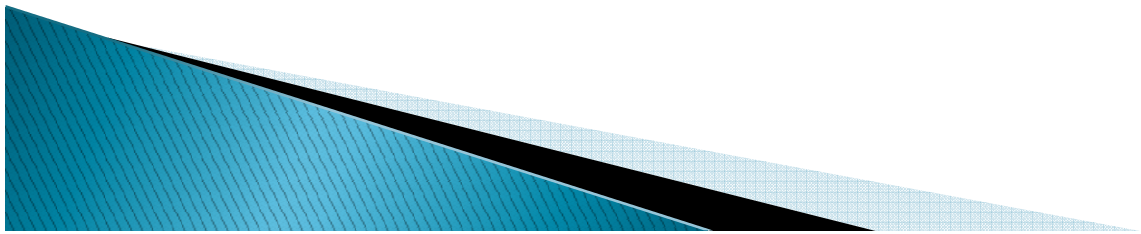
# Le paradigme Objet

- Programmation orientée objet
  - Méthode d'implantation dans laquelle le programme est organisé en collection d'objets coopératifs. Chaque objet est une instance de classe. Toutes les classes sont membres d'une hiérarchie de classes liées pas des relations d'héritage
- Conception orientée objet
  - Méthode de conception qui mène à une décomposition orientée objet
- Analyse orientée objet
  - Méthode d'analyse qui examine les besoins en termes de classes et d'objets trouvé dans l'espace du problème



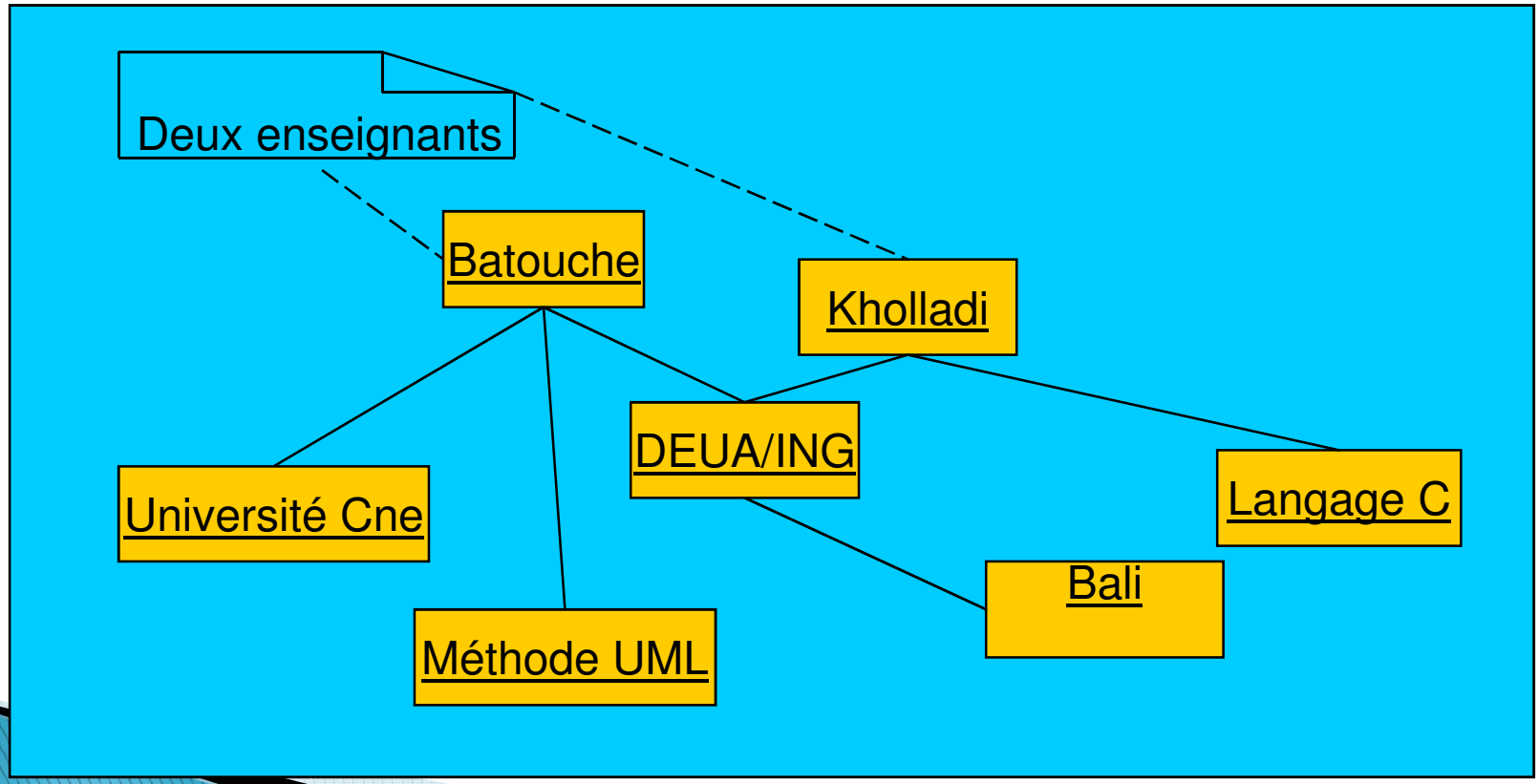
# Objets

- Définitions
  - Objet informatique :
    - Est une unité atomique formée de l'union d'un état et d'un comportement
    - Définit une représentation abstraite d'une entité du monde réel ou virtuel, dans le but de la piloter ou de la simuler
      - Grain de sable, étoile
      - Compte en banque, police d'assurance, équation mathématiques, etc.
  - Les objets du monde informatique naissent, vivent et meurent



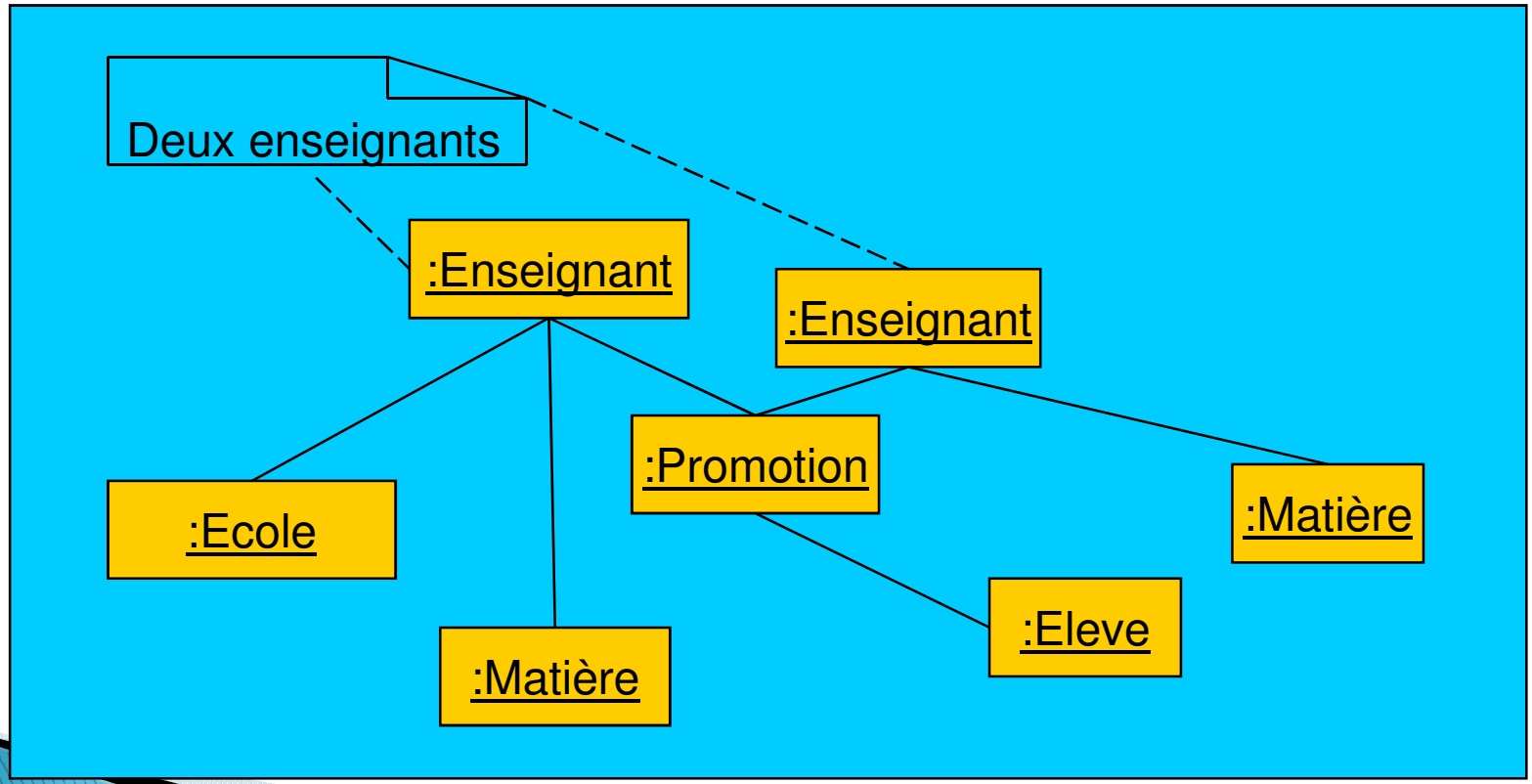
# Objets

Représentation d'objets en UML



# Objets

- Représentation d'objets en UML



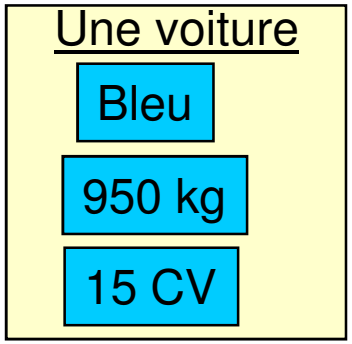
# Objets – Caractéristiques fondamentales

- Caractéristique fondamentales d'un objet (informatique)

  Objet = État + Comportement + Identité

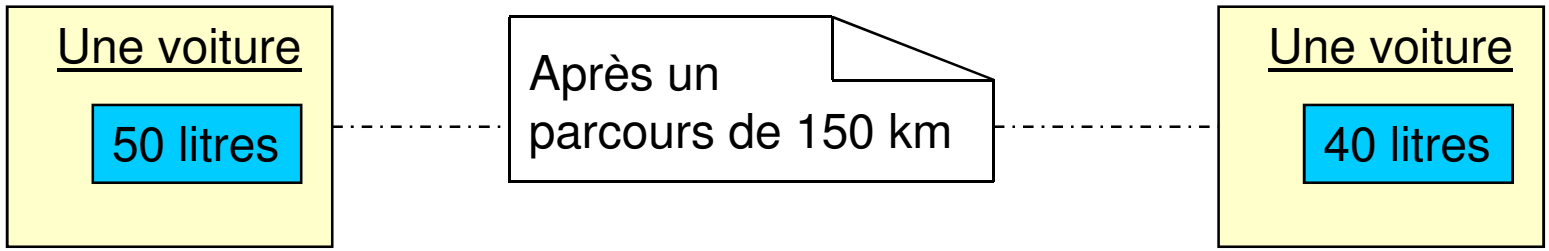
  - État
    - Regroupe les valeurs instantanées de tous les attributs d'un objet :
      - Un attribut est une information qui qualifie l'objet qui le contient
      - Chaque attribut peut prendre une valeur dans un domaine de définition donné

    - **Exemple** : Un objet *voiture* regroupe les valeurs des attributs *couleur*, *masse* et *puissance fiscale*



# Objets – Caractéristiques fondamentales

- État
  - L'état d'un objet à un instant donné correspond à une sélection de valeurs, parmi toutes les valeurs possibles des différent attributs
  - L'état évolue au cours du temps,  il est la conséquence de ses comportement passés
    - *Une voiture roule, la quantité de carburant diminue, les pneus s'usent, etc.*

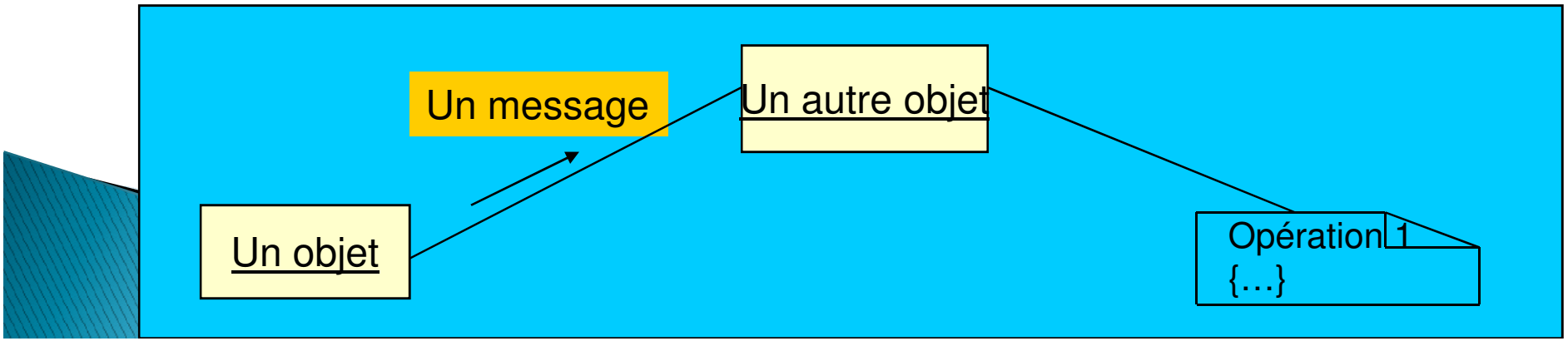



**REMARQUE :** Certaines composantes de l'état peuvent être constantes

La marque de la voiture, pays de la construction de la voiture

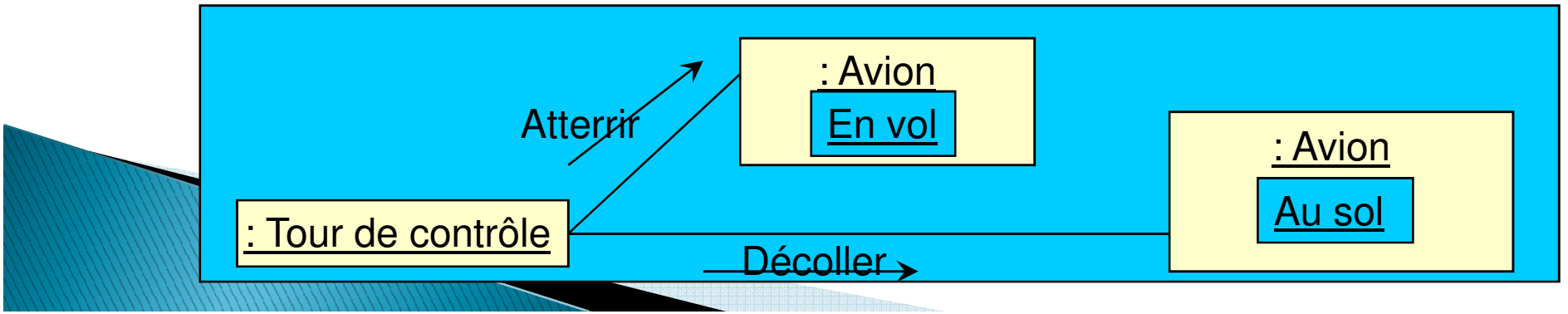
# Objets – Caractéristiques fondamentales

- Comportement
  - Regroupe toutes les compétences d'un objet et décrit les actions et les réactions de cet objet
  - Chaque atome (partie) de comportement est appelé opération
    - Les opérations d'un objet sont déclenchées suite à une stimulation externe, représentée sous la forme d'un message envoyé par un autre objet
    - L'état et le comportement sont liés



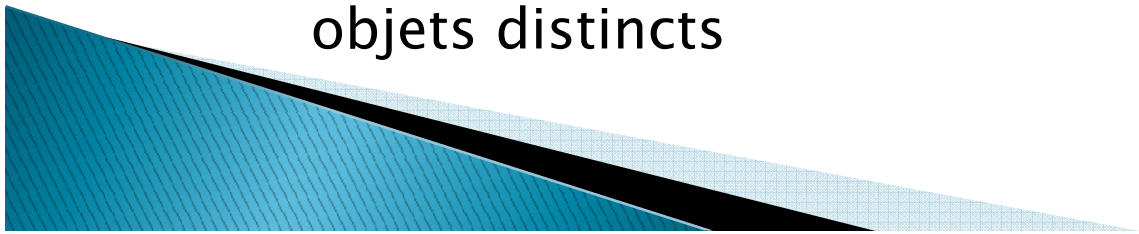
# Objets – Caractéristiques fondamentales

- Le comportement
  - Le comportement à un instant donné dépend de l'état courant et l'état peut être modifié par le comportement
    - **Exemple** : il n'est possible de faire atterrir un avion que s'il est en train de voler: le comportement *Atterrir* n'est valide que si l'information *En vol* est valide
    - Après l'atterrissage, l'information *En vol* devient invalide et l'opération *Atterrir* n'a plus de sens



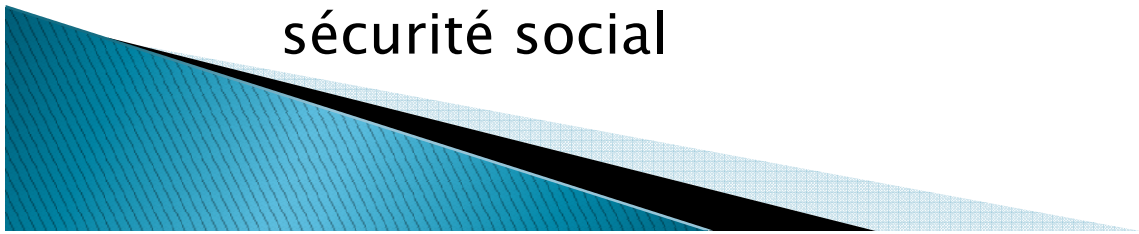
# Objets – Caractéristiques fondamentales

- Identité
  - Chaque objet possède une identité qui caractérise son existence propre
  - L'identité permet de distinguer tout objet de façon non ambiguë, indépendamment de son état
  - Permet de distinguer deux objets dont toutes les valeurs d'attributs sont identiques
    - deux pommes de la même couleur, du même poids et de la même taille sont deux objets distincts
    - Deux véhicules de la même marque, de la même série et ayant exactement les mêmes options sont aussi deux objets distincts

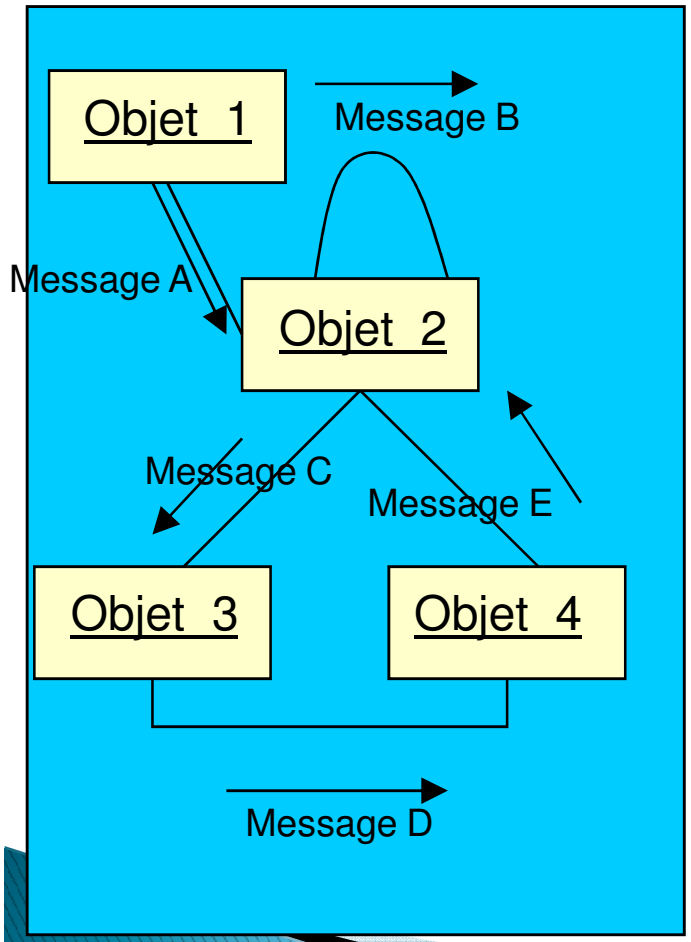


# Objets – Caractéristiques fondamentales

- Identité
  - L'identité est un concept, elle ne se représente pas de manière spécifique en modélisation : chaque objet possède une identité de manière implicite
  - En phase de réalisation, l'identité est souvent construite à partir d'un identifiant issu naturellement du domaine du problème
    - Nos voitures possèdent toutes un numéro d'immatriculation, nos téléphones un numéro d'appel et nous-mêmes sommes identifiés par nos numéros de sécurité social



# Objets – Communication entre objets

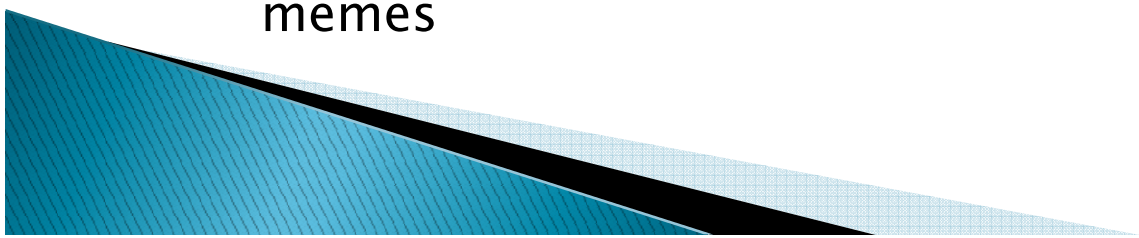



- **Le concept de message**
  - Les systèmes informatiques à objets peuvent être vus comme des sociétés d'objets qui travaillent en synergie afin de réaliser les fonctions de l'application
  - Le comportement global d'une application repose sur la communication entre les objets qui la composent
  - L'unité de communication entre objets s'appelle **message**
  - Il existe **cinq catégories** principales de messages
    - Les constructeurs qui créent des objets
    - Les destructeurs qui détruisent des objets
    - Les sélecteurs qui renvoient tout ou partie de l'état de l'objet
    - Les modificateurs qui changent tout ou partie de l'état d'un objet
    - Les itérateurs qui visitent l'état d'un objet

# Classes

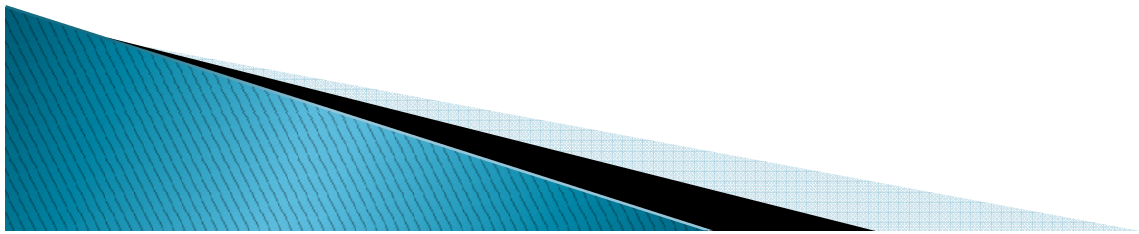
- Définition
  - Une classe décrit une abstraction d'objets ayant :
    - Des propriétés similaires
    - Un comportement commun
    - Des relations identiques avec les autres objets
    - Une sémantique commune
  - Par exemple *Véhicule* est la classe de tous les véhicules

- Une classe a trois fonctions:
  - Sert de « patron » (*template*) à objets : elle définit la structure générale des objets qu'elle
  - Sert de « conteneur » d'objets : contient l'ensemble des objets qu'elle a créés
  - Sert de « réceptacle » des *méthodes* que ses objets peuvent utiliser puisque tous les objets d'une classe utilisent les mêmes *méthodes* , il serait inutile de les dupliquer dans ces objets eux-mêmes



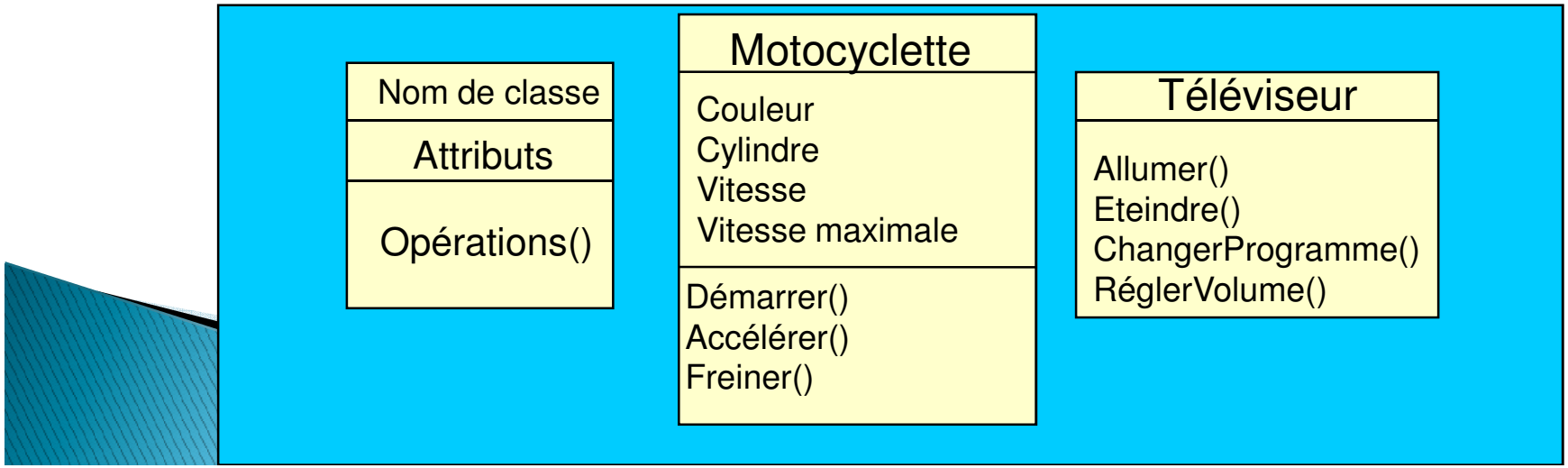
# Classes

- Caractéristiques d'une classe
  - Un objet créé par (ou appartenant à) une classe sera appelé une *instance de cette classe*
  - les valeurs des *variables d'instances* sont propres à chacune de ces instances et les caractérisent
  - Les généralités sont contenues dans la classe et les particularités sont contenues dans les objets
  - Les objets sont construits à partir de la classe, par un processus appelé instanciation : tout objet est une instance de classe
  - Nous distinguons deux types de classes
    - Classe concrète : peut être instanciée
    - Classe abstraite : est une classe qui ne donne pas directement des objets



# Classes – Représentation graphique

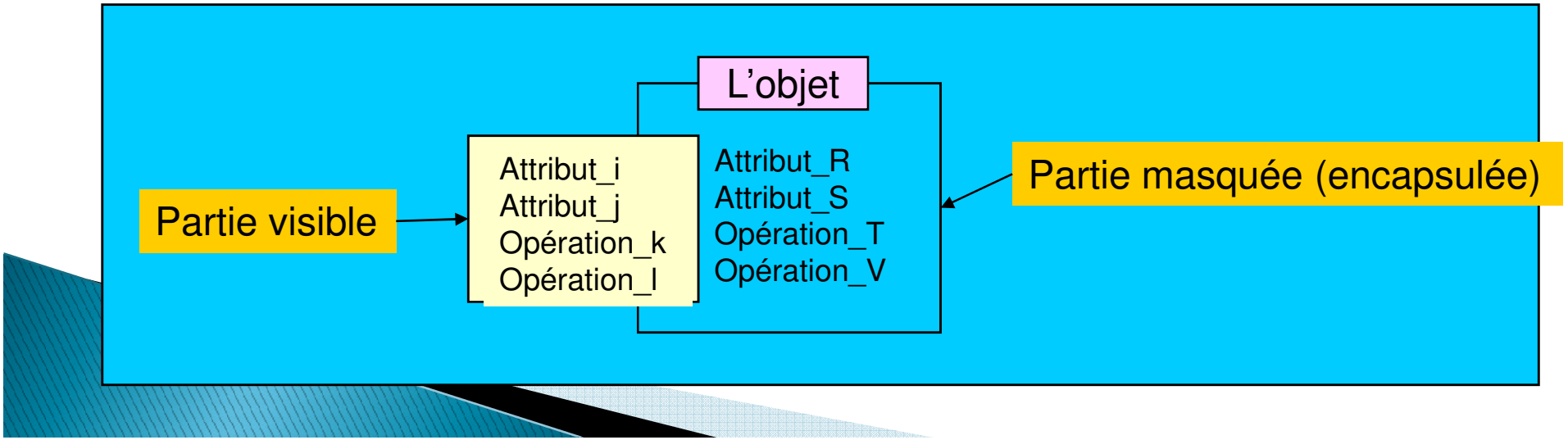
- Chaque classe est représentée sous la forme d'un rectangle divisé en trois compartiments
- Les compartiments peuvent être supprimés pour alléger les diagrammes
- Représentation des classes abstraites : le nom d'une classe abstraite est en italique



# Encapsulation

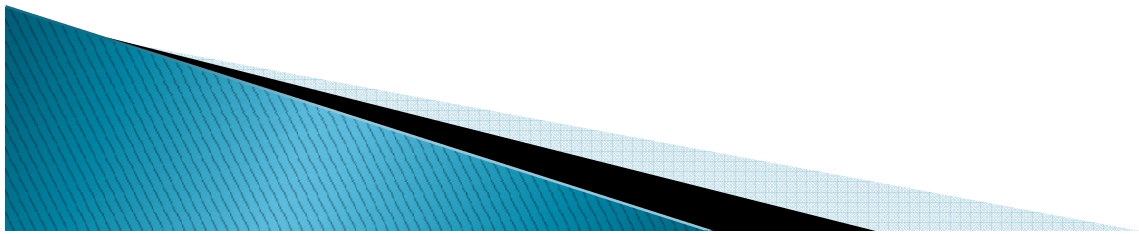
- Consiste à masquer les détails d'implémentation d'un objet, en définissant une interface
  - séparation entre les propriétés externes, visibles des autres objets, et les aspects internes, propres aux choix d'implantation d'un objet

- L'interface est la vue externe d'un objet, elle définit les services accessibles (offerts) aux utilisateurs de l'objet

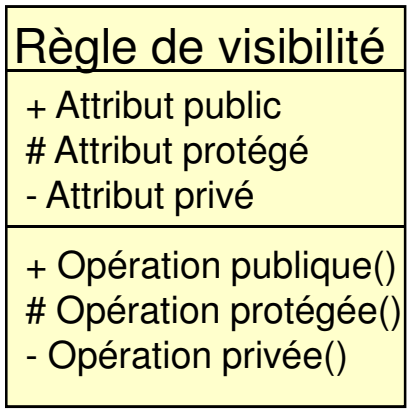


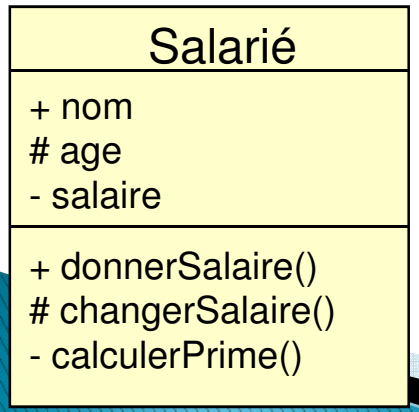
# Encapsulation

- Avantages
  - Facilite l'évolution d'une application car elle stabilise l'utilisation des objets : on peut modifier l'implémentation des attributs d'un objet sans modifier son interface
    - Les utilisateurs d'une abstraction ne dépendent pas de la réalisation de l'abstraction mais seulement de sa spécification : ce qui réduit le couplage dans les modèles
  - Garantit l'intégrité des données, car elle permet d'interdire l'accès direct aux attributs des objets (utilisation d'accesseurs)



# Encapsulation

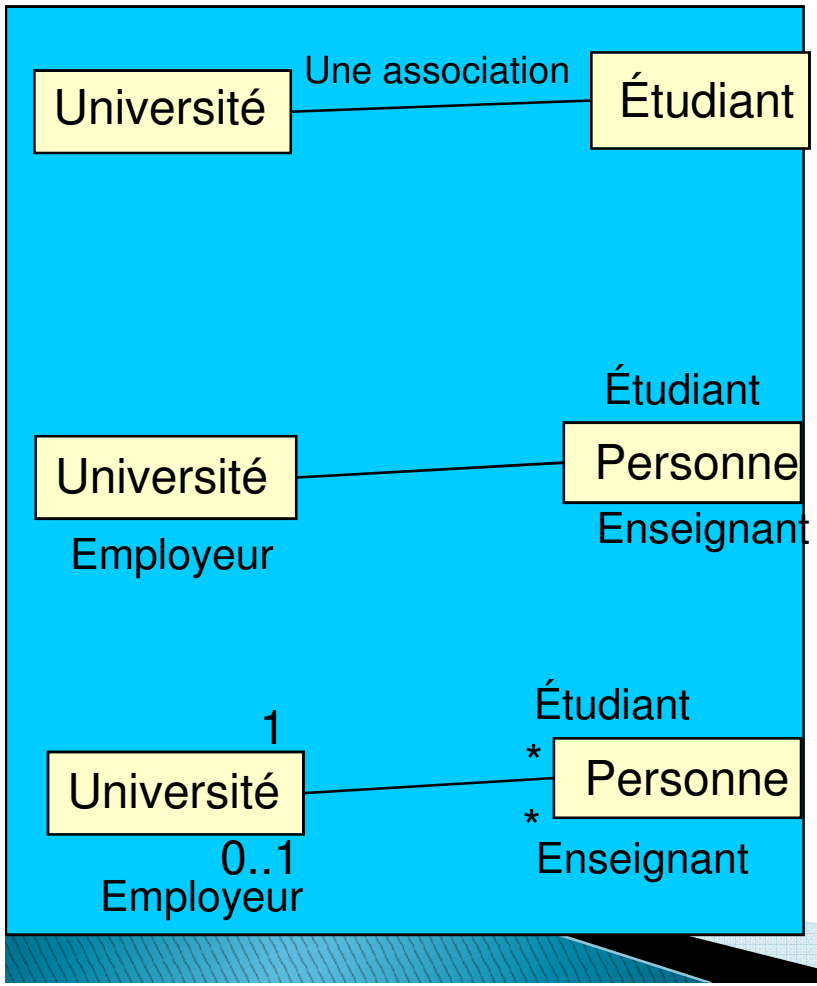



- Il est possible d'assouplir le degré d'encapsulation au profit de certaines classes utilisatrices bien particulières
  - En définissant des niveaux de visibilité

- Les trois niveaux distincts d'encapsulation
  - Niveau privé : c'est le niveau le plus fort; la partie privée de la classe est totalement opaque
  - Niveau protégé : c'est le niveau intermédiaire. Les attributs placés dans la partie protégée sont visibles par les classes dérivées de la classe fournisseur. Pour toutes les autres classes, les attributs restent invisibles
  - Niveau public : ceci revient à se passer de la notion d'encapsulation et de rendre visibles les attributs pour toutes les classes

# Les relations entre les classes




- Associations
  - Représentent les relations structurelles qui existent entre objets de différentes classes.
  - La plupart des associations sont binaires
  - Une association est caractérisée par:
    - un nom, qui peut être omis notamment quand les rôles des classes sont spécifiés
    - un rôle pour chacune des classes qui participent à l'association. Le rôle d'une classe, écrit à l'extrémité du lien d'association, décrit la manière dont cette classe est « vue » par l'autre classe.
    - une multiplicité

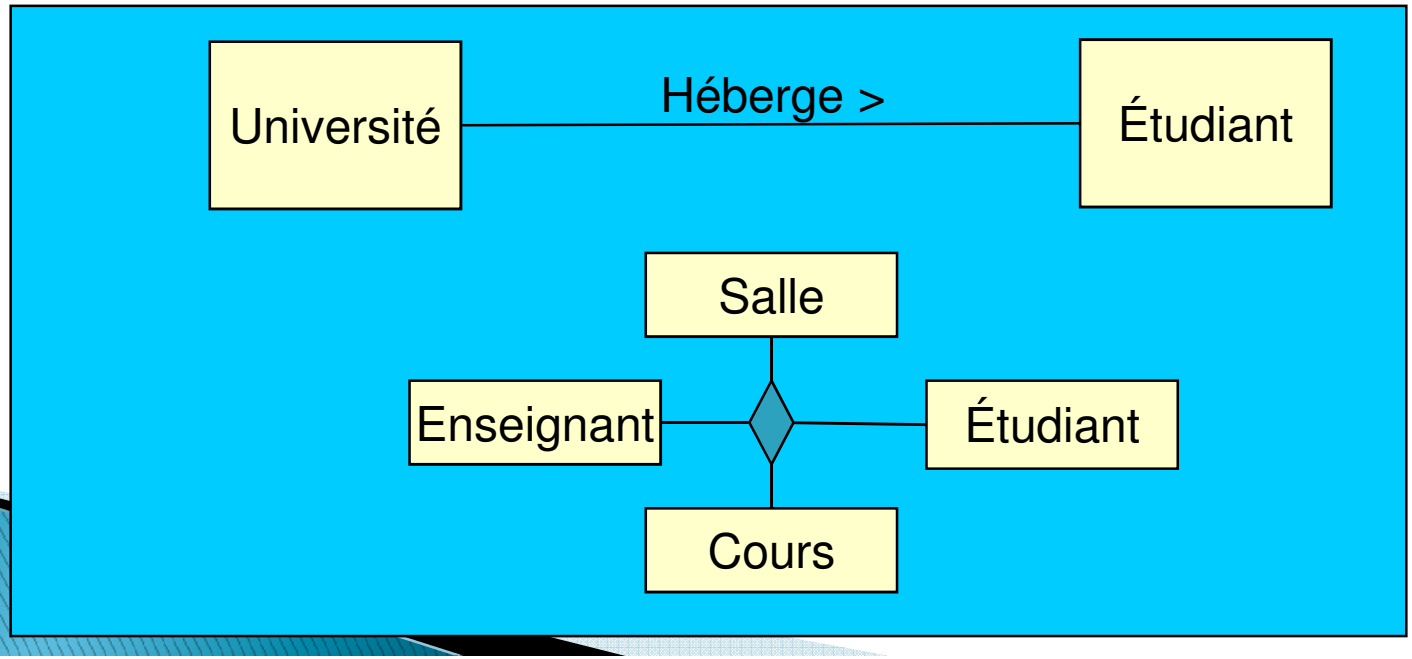
# Les relations entre les classes

- Multiplicité
  - Chaque extrémité d'une association peut porter une indication de multiplicité (nombre d'occurrences) qui montre combien d'objets de la classe considérée peuvent être liés à un objet de l'autre classe.

| | |
|---|---|
| 1 | Un et un seul |
| 0..1 | Zéro ou un |
| M..N | De M à N (entiers naturels) |
| * | De zéro à plusieurs |
| 0..* | De zéro à plusieurs |
| 1..* | D'un à plusieurs |

# Les relations entre les classes

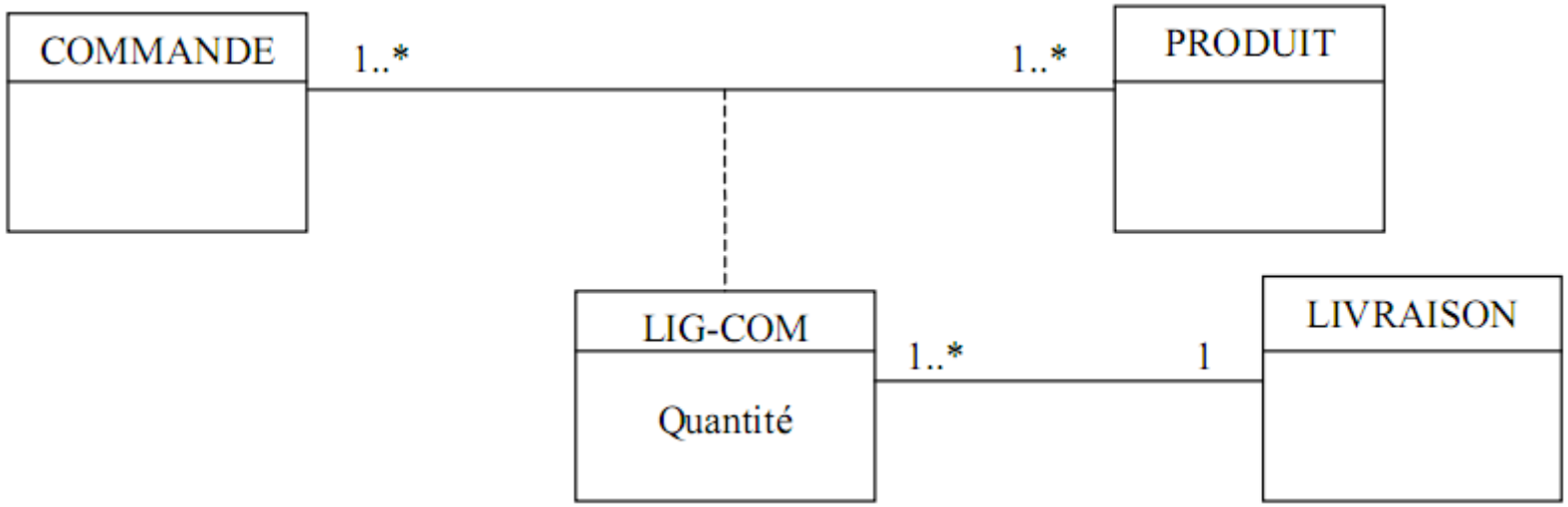
- Le nom de l'association peut être suivi d'une flèche de direction (triangle plein) qui précise le sens de lecture
- Il existe des associations ternaires ou d'ordre supérieur qui sont représentées par un losange relié à chacune des classes participant à la relation.



# Les relations entre les classes

- Les classes associations
  - Il peut arriver que l'on ait besoin de garder des informations (attributs ou opérations) propres à une association. Une classe de ce type est appelée classe association.

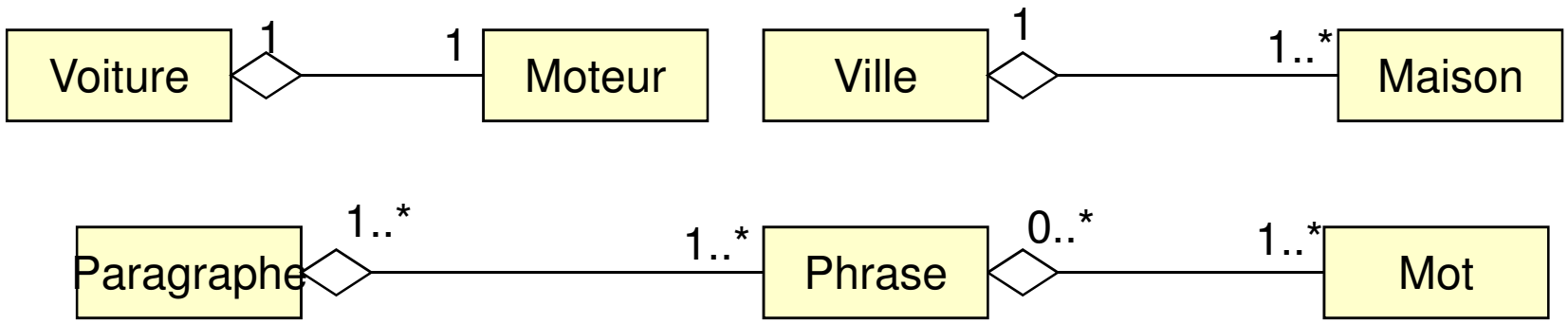
  - **Remarque** : une classe association peut entretenir des relations avec d'autres classes



# Les relations entre les classes

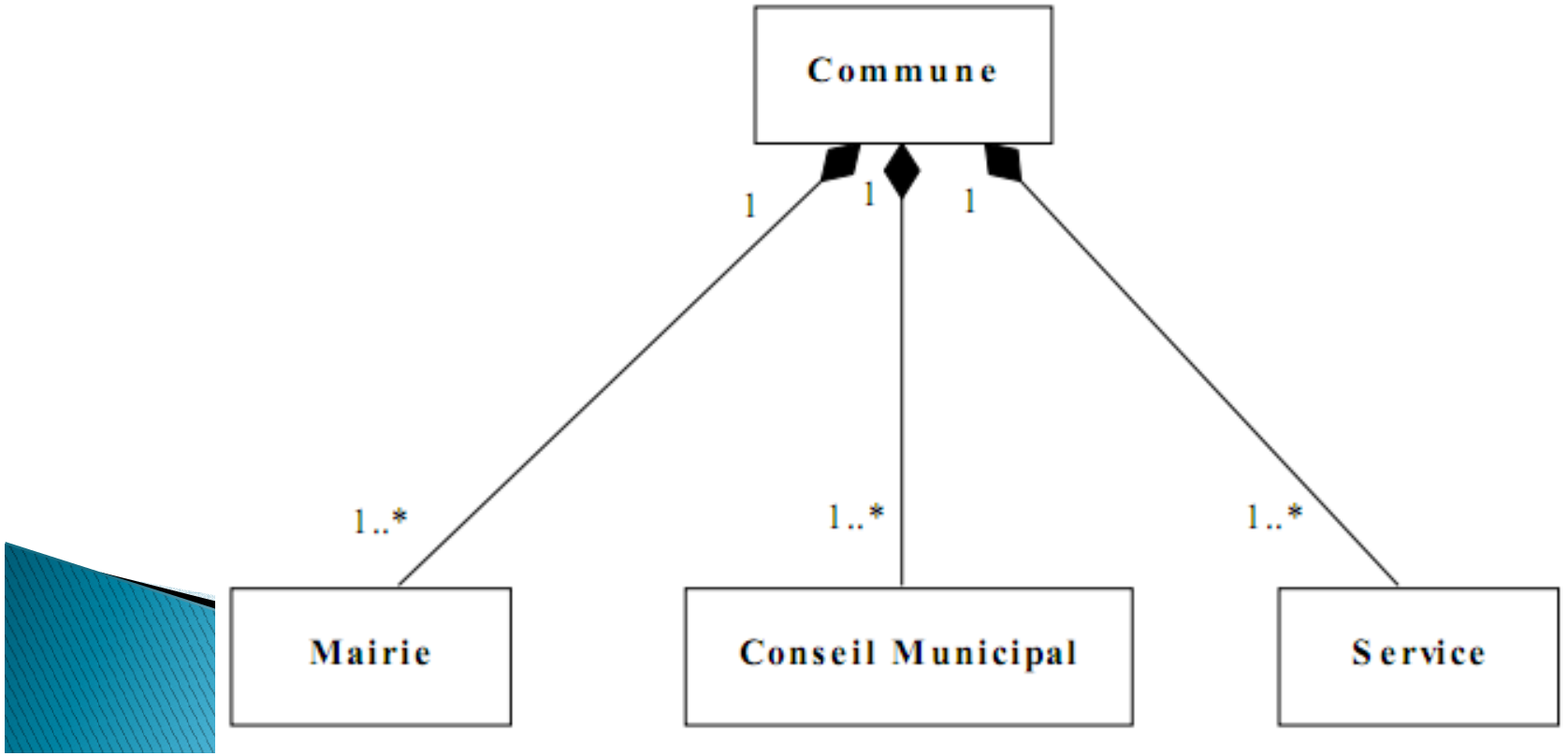
- L'agrégation
  - Est une forme particulière d'association
  - Permet de renforcer la dépendance entre classes
  - Est une association non symétrique dans laquelle une des extrémités joue un rôle prédominant par rapport à l'autre extrémité

  - Les critères suivants impliquent une agrégation :
    - une classe fait partie d'une autre classe,
    - une action sur une classe implique une action sur une autre classe,
    - les objets d'une classe sont subordonnés aux objets d'une autre classe.



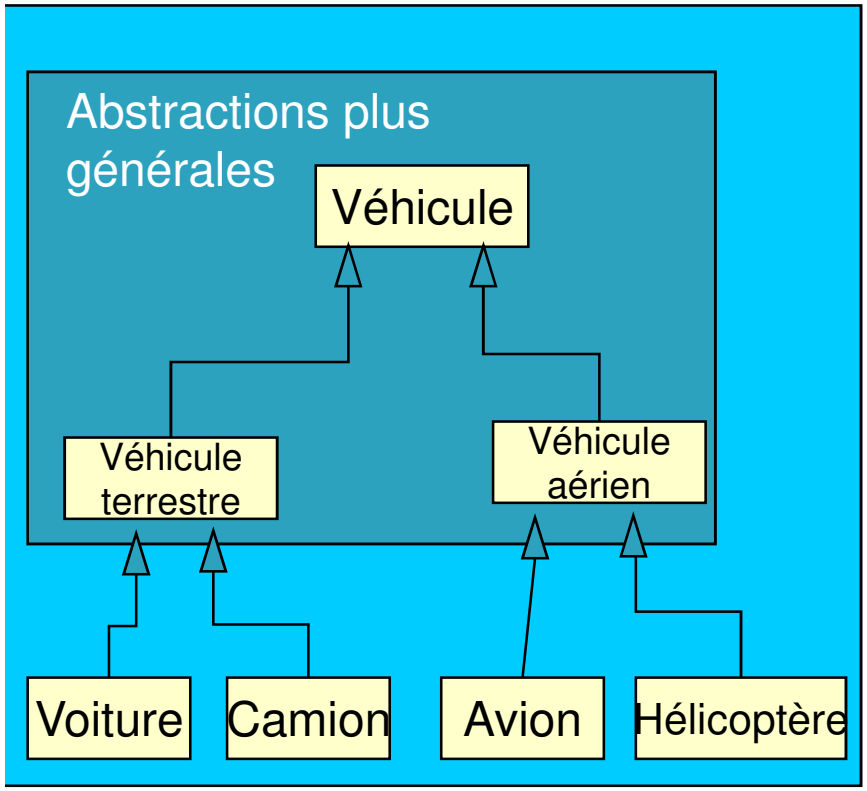
# Les relations entre les classes

- La composition
  - Cas particulier d'agrégation dans laquelle la vie des composants est liée à celle de l'agrégat (association « *estPartieDe* »)



# Les hiérarchies de classes

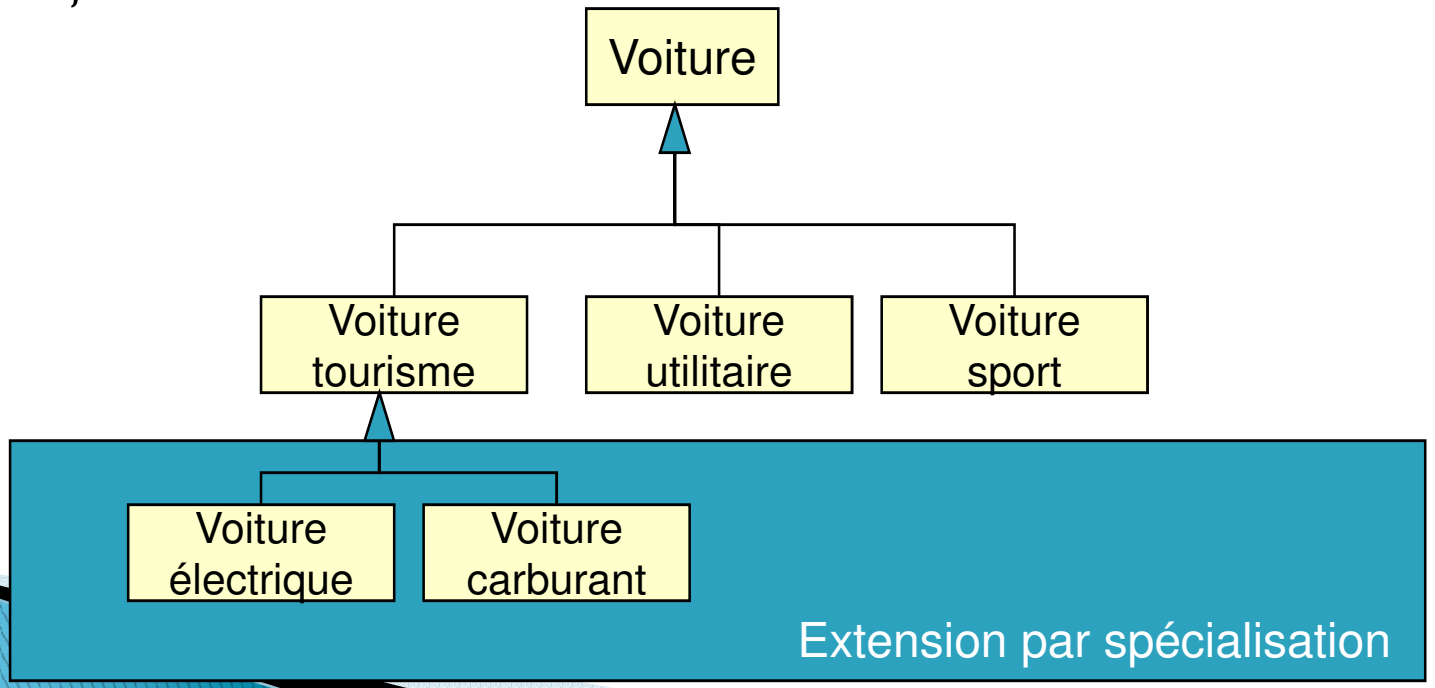
- Généralisation et spécialisation




- La généralisation consiste à factoriser les éléments communs (attributs, opérations) d'un ensemble de classes dans une classe plus générale appelée super-classe
  - Les classes sont ordonnées selon une hiérarchie ; une super-classe est une abstraction de ses sous-classes

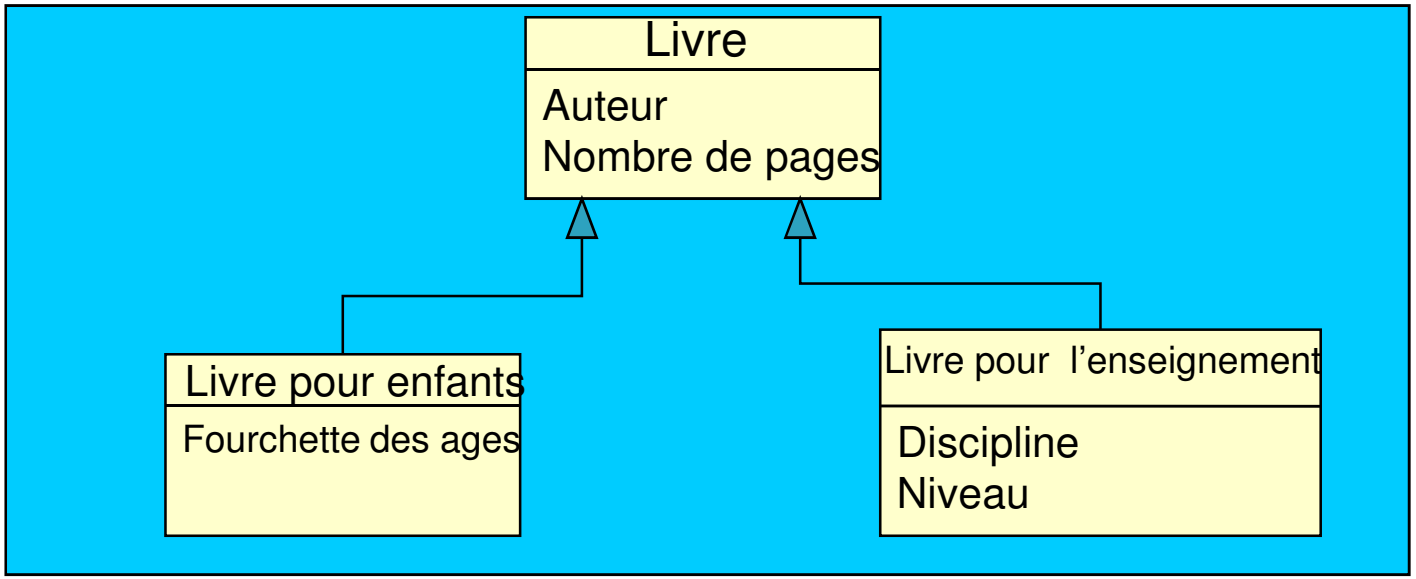
# Les hiérarchies de classes

- La spécialisation permet de capturer les particularités d'un ensemble d'objets non discriminées par les classes déjà identifiées
  - Les nouvelles caractéristiques sont représentés par une nouvelle classe, sous-classe d'une des classes existantes



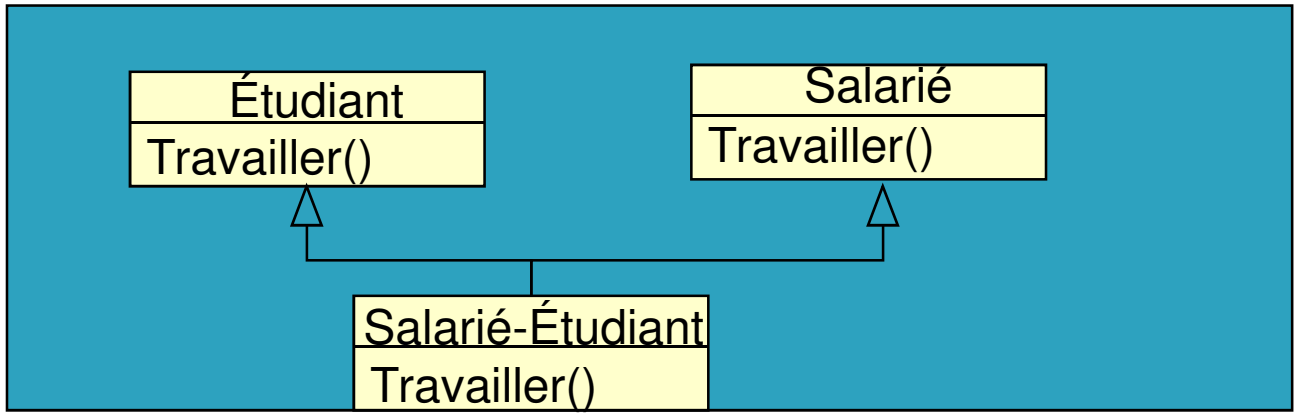
# Les hiérarchies de classes

- Les propriétés d'une sous-classe englobent les propriétés de toutes ses super-classes



# Les hiérarchies de classes

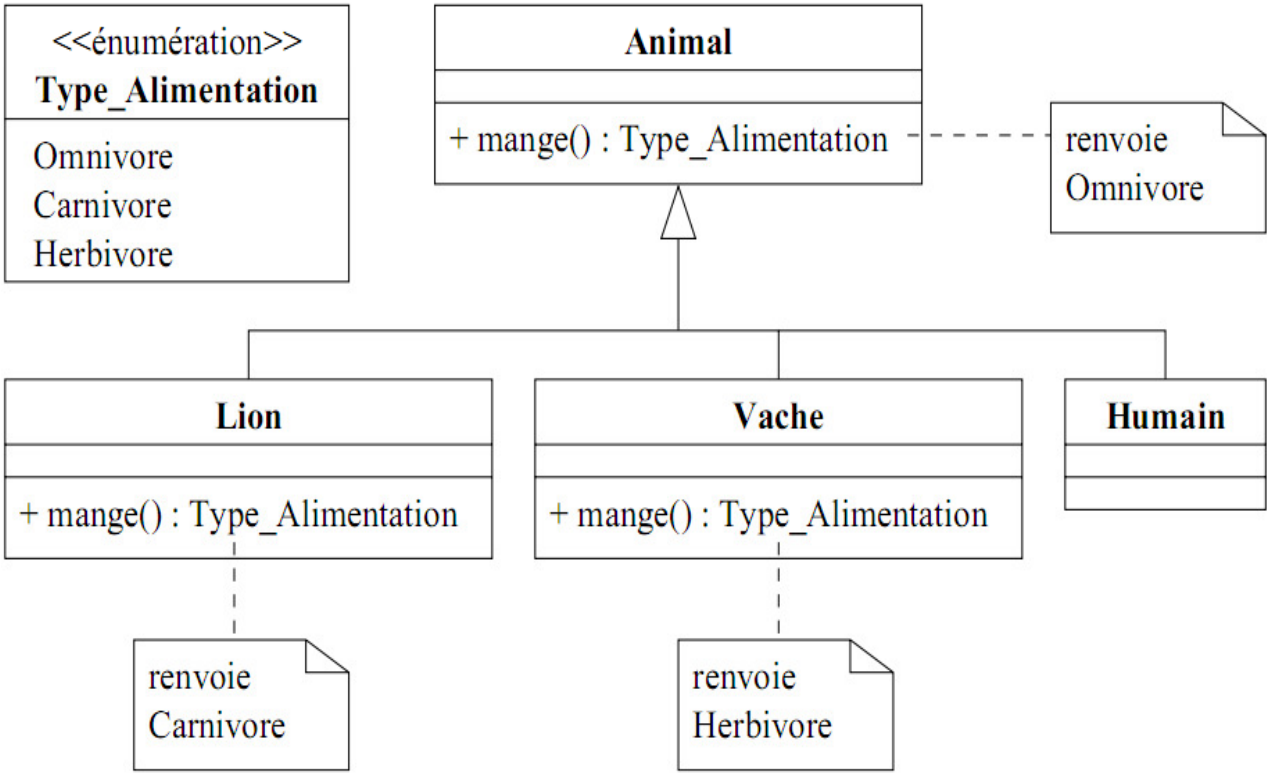
- Généralisation multiple
  - Une classe n peut posséder qu'une fois une propriété donné




- Problème de classification
  - L'établissement d'une hiérarchie (classification) dépend du point de vue
    - Pas une seule hiérarchie de classes mais des hiérarchies, chacune adaptée à un usage donné
    - **Exemple** : les animaux
      - Critères de classification : type de nourriture, la protection des animaux

# Les hiérarchies de classes

- Le polymorphisme
  - Le terme polymorphisme décrit la caractéristique d'un élément qui peut prendre plusieurs formes, comme l'eau qui se trouve à l'état solide, liquide ou gazeux
  - **Définition** : mécanisme permettant à une classe fille la spécialisation d'opérations



# Exercices